IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Biran et al.	**Conf. No.**:	8413
Serial No.:	10/733,734	**Art Unit**:	2146
Filed:	12/11/2003	**Examiner**:	Musa
Title:	RDMA NETWORK INTERFACE CONTROLLER WITH CUT-THROUGH IMPLEMENTATION FOR ALIGNED DDP SEGMENTS BASED ON VALIDITY OF CYCLICAL REDUNDANCY CHECK	**Docket No.**:	FIS920030290US1 (IBMF-0039)

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF OF APPELLANTS

This is an appeal from the Final Rejection (OA) dated May 5, 2008, rejecting claims 1-40. The requisite fee set forth in 37 C.F.R. §1.17 (c) has been submitted on July 8, 2008.

REAL PARTY IN INTEREST

International Business Machines Corporation is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

As filed, this case includes claims 1-40. Claims 1-40 remain pending, among which claims 1-40 stand rejected, and form the basis of this appeal. No claim has been allowed. The rejections of claims 1-40 are being appealed.

STATUS OF AMENDMENTS

An amendment has been filed following the Final Rejection of May 5, 2008. This amendment was filed to correct typographical errors in claims 20, 22-31, 35, 36 and 38-40. In the second Advisory Action mailed August 5, 2008, the Office indicated that the amendment would be entered for the purposes of appeal.

SUMMARY OF THE CLAIMED SUBJECT MATTER

The present invention as defined by independent claim 1 is a method of handling a data transfer in a network interface controller (NIC). The method includes receiving the data transfer wherein the data transfer is denoted as one of a first type and a second type (page 15, lines 2-3). A cyclical redundancy check (CRC) for the data transfer is calculated, wherein the CRC is one of valid and invalid (page 15, lines 4-5). Based on a comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length and validity of a CRC of a direct data placement (DDP) segment within the data transfer one of the following is conducted (page 20, paragraph [0055]); 1) dropping the data transfer and not confirming reception; 2) placing the data transfer to a reassembly buffer of the NIC; and 3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer (page 15, lines 5-7).

The present invention, defined by claim independent claim 19, is a network interface controller (NIC) for handling a data transfer. The NIC includes a first storage means for storing the data transfer for reassembly (page 15, lines 9 and 10) and a second storage means for storing the data transfer for direct data placement to a destination buffer (page 15, lines 10 and 11). There are means provided for receiving the data transfer wherein the data transfer is denoted as one of a first type and a second type (page 15, lines 11 and 12) and means for calculating a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid (page 15, lines 12 and 13). There are means for, based on a comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length and validity of a CRC of a direct data placement (DDP) segment within the data transfer (page 20, paragraph [0055]), conducting one of: 1) dropping the data transfer and not confirming reception (page 15, line 14); 2) placing the data transfer to a reassembly buffer of the NIC (page 15, lines 14 and 15); and 3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer (page 15, lines 16 and 17).

The present invention, as defined by claim 37 is a computer program product comprising a tangible computer useable medium having computer readable program code embodied therein (page 15, line 17-18), which, when executed by a computer infrastructure, enables the computer infrastructure to handle a data transfer in a network interface controller (NIC) (page 15, line 19), the program product including program code configured to receive the data transfer wherein the data transfer is denoted as one of a first type and a second type (page 15, lines 20 and 21), program code configured to calculate a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid (page 15, lines 21-23), program code configured to conduct, based on a comparison between a transfer control protocol (TCP) segment length and a

marker with protocol data unit alignment (MPA) length and validity of a CRC of a direct data placement (DDP) segment within the data transfer (page 20, paragraph [0055]), one of; 1) dropping the data transfer and not confirming reception (page 16, lines 1-2); 2) placing the data transfer to a reassembly buffer of the NIC (page 16, line 2); and 3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer (page 16, lines 2-3).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-40 are unpatentable under 35 USC 103(a) over Craft et al. (US 7,124,205), hereinafter "Craft" in view of Starr et al. (US Pub. No. 2004/0064590 A1), hereinafter "Starr."

ARGUMENT

1. Claims 1-40 are not obvious over Craft in view of Starr.

With respect to independent claims 1, 19 and 37, Craft does not disclose "dropping the data transfer and not confirming reception". The Office points to col. 23, line 8 and col. 40, line 50 of Craft; however, a careful reading of those passages reveals that rather than dropping the data transfer and not confirming reception, the connection is "flushed" back to the host computer for slow-path processing (col. 40, lines 50-57). Starr is cited by the Office to show the message packet sent to the host undergoes cyclical redundancy checking and is then sent across an I/O bus and stored in the host memory and also for teaching comparators that compare data with a corresponding signal to match memory information to output valid information. However, Starr does not correct the deficiency of Craft cited above, namely "dropping the data transfer and not confirming reception". In fact Starr, teaches flushing back to the host computer [paragraph

0059]. Thus, neither Craft nor Starr discloses an essential element of Appellants claims.

In the Advisory Action, the element of "dropping the data transfer and not confirming reception" in the claims is not addressed by the Office. The citation of col. 39-42 of Craft does not show this element. As noted above, Craft teaches flushing back to the host computer for slow-path processing (col. 40, lines 50-57). The definition of flushing a connection (Craft, col. 40, lines 55-63) is defined as making the communication control block on the NIC invalid and making the communication control block on the host computer valid. Fast-path processing of the packets for the flushed connection are passed to the protocol stack of the host computer for slow-path processing. Error handling is done by the protocol stack. This is further described in column 41, lines 12-22 of Craft, wherein the NIC when flushing a connection is sending an error message to the host computer to determine the part of the message that remains to be filled. Thus, rather than dropping the data transfer and not confirming reception, as required in Appellants claims, Craft teaches labeling the communication block invalid and sending an error message to the host computer. This is not dropping data transfer and not confirming reception. Therefore, a *prima facia* case of obviousness has not been made.

In addition, a close reading of Starr reveals a similar teaching. In the case where the packet summary of a communication block contains exception conditions or errors, the communication block and packet are flushed to the host protocol stack for protocol processing. If the destination for the packet is not indicated, the packet is sent to the host protocol stack to determine the destination [paragraph 0059 of Starr]. Thus, there is no teaching of dropping data transfer and not confirming reception. Indeed, Starr teaches against this element, as whenever there is an error in the communication between the NIC and host, communication is always maintained in an attempt to determine the error [paragraph 0059 of Starr]. This is the same

teaching as Craft, i.e. the connection is never dropped, rather there are different procedures for each type of error but in no case is the connection dropped without confirming reception. In summary, neither Craft nor Starr teaches the element of dropping the data transfer and not confirming reception. Moreover both Craft and Starr teach against this element as they both flush back to the host computer to determine the error and resend the packet. For these reasons, Appellants request withdrawal of the rejections.

Should the Office believe that anything further is necessary to place the application in better condition for allowance, the Office is requested to contact Appellants' undersigned attorney at the telephone number listed below.

Respectfully submitted,

/Carl F. Ruoff/

Carl F. Ruoff
Reg. No. 34,241

Dated: August 19, 2008

Hoffman Warnick LLC
75 State Street, 14th Floor
Albany, New York 12207
(518) 449-0044
(518) 449-0047 (fax)

CLAIMS APPENDIX

1. A method of handling a data transfer in a network interface controller (NIC), the method comprising:

a) receiving the data transfer wherein the data transfer is denoted as one of a first type and a second type;

b) calculating a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid; and

c) based on a comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length and validity of a CRC of a direct data placement (DDP) segment within the data transfer, conducting one of:

1) dropping the data transfer and not confirming reception;

2) placing the data transfer to a reassembly buffer of the NIC; and

3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer.

2. The method of claim 1, wherein c), 2) is conducted in the case that the data transfer is of the first type.

3. The method of claim 1, further comprising determining whether the data transfer includes a single or multiple direct data placement (DDP) segments.

4. The method of claim 3, wherein c), 3) is conducted in the case that the data transfer includes multiple DDP segments and all DDP segments have a valid CRC that is fully contained

in a TCP segment.

5. The method of claim 3, wherein c), 1) is conducted in the case that the data transfer includes multiple DDP segments, a first DDP segment has an invalid CRC, and a DDP header of the first DDP segment is referred by an MPA length associated with a previous DDP segment.

6. The method of claim 5, wherein, in the case that the data transfer includes multiple DDP segments, a first DDP segment has an invalid CRC, and the DDP header of the first DDP segment is not referred by the MPA length associated with the previous DDP segment:

c), 1) is conducted in the case that the DDP header is referred by an MPA marker; and

c), 2) is conducted in the case that the DDP header is not referred by the MPA marker.

7. The method of claim 3, wherein c), 1) is conducted in the case that the data transfer includes multiple DDP segments and a last DDP segment extends outside of the TCP segment boundary;

and c), 2) is conducted in the case that the data transfer includes multiple DDP segments and a last DDP segment does not extend outside of the TCP segment boundary.

8. The method of claim 2, wherein c), 2) is conducted in the case that the data transfer includes a single DDP segment and an MPA length associated with the single DDP segment is greater than a transmission control protocol (TCP) segment length of the data transfer.

9. The method of claim 2, wherein c), 3) is conducted in the case that the data transfer

includes a single DDP segment that has: an MPA length associated therewith that equals a TCP segment length and a valid CRC.

10. The method of claim 2, wherein c), 1) is conducted in the case that the data transfer includes a single DDP segment that has: an MPA length associated therewith that equals a TCP segment length, an invalid CRC and a DDP header that is referred by an MPA length associated with a previous DDP segment.

11. The method of claim 2, wherein in the case that the data transfer includes a single DDP segment that has: an MPA length associated therewith that equals a TCP segment length, an invalid CRC and a DDP header that is not referred by an MPA length associated with a previous DDP segment:

c), 1) is conducted in the case that the DDP header is referred by an MPA marker; and

c), 2) is conducted in the case that the DDP header is not referred by an MPA marker.

12. The method of claim 1, further comprising setting the data transfer type to the first type when c), 2) is conducted.

13. The method of claim 1, wherein in the case that c), 3) is conducted on an out-of-order data transfer, the method further comprises:

clearing TCP hole information created by the out-of-order data transfer in a connection context; and

stopping receipt reporting for the out-of-order data transfer.

14. The method of claim 1, wherein the data transfer includes DDP segments, and the calculating includes calculating a CRC for all DDP segments of the data transfer together.

15. The method of claim 14, wherein the data transfer does not contain an MPA marker.

16. The method of claim 14, further comprising:

storing a number of retransmission attempts for each data transfer including an error; and

storing a largest sequence number.

17. The method of claim 16, wherein in the case that CRC is invalid for the data transfer, which indicates the data transfer is a newly received error-including data transfer:

c), 2) is conducted on the newly received error-including data transfer in the case that the number of retransmission attempts exceeds a maximum retransmission attempt number for that data transfer, and

c), 1) is conducted on the newly received error-including data transfer in the case that the number of retransmission attempts does not exceed a maximum retransmission attempt number for that data transfer; and

wherein in the case that c), 1) is conducted, the method further comprises:

increasing the number of retransmission attempts for the newly received error-including data transfer by one; and

updating the largest sequence number to carry the largest sequence number among at least one previously received error-including data transfer and the newly

received error-including data transfer.

18. The method of claim 16, wherein in the case that CRC is valid for an in-order data transfer:

a) in the case that a sequence number of the in-order data transfer is greater than the stored largest sequence number, the number of retransmission attempts is reset and c), 3) is conducted; and

b) in the case that the sequence number of the in-order data transfer is not greater than the stored largest sequence number, c), 3) is conducted.

19. A network interface controller (NIC) for handling a data transfer, the NIC comprising:

first storage means for storing the data transfer for reassembly;

second storage means for storing the data transfer for direct data placement to a destination buffer;

means for receiving the data transfer wherein the data transfer is denoted as one of a first type and a second type;

means for calculating a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid; and

means for, based on a comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length and validity of a CRC of a direct data placement (DDP) segment within the data transfer, conducting one of:

1) dropping the data transfer and not confirming reception;

2) placing the data transfer to a reassembly buffer of the NIC; and

3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer.

20. The NIC of claim 19, wherein the conducting means conducts 2) in the case that the data transfer is of the first type.

21. The NIC of claim 19, further comprising means for determining whether the data transfer includes a single or multiple direct data placement (DDP) segments.

22. The NIC of claim 21, wherein the conducting means conducts 3) in the case that the data transfer includes multiple DDP segments and all DDP segments have a valid CRC that are fully contained in TCP segment.

23. The NIC of claim 21, wherein the conducting means conducts 1) in the case that the data transfer includes multiple DDP segments, a first DDP segment has an invalid CRC, and a DDP header of the first DDP segment is referred by an MPA length associated with a previous DDP segment.

24. The NIC of claim 21, wherein in the case that the data transfer includes multiple DDP segments, a first DDP segment has an invalid CRC, and a DDP header of the first DDP segment is not referred by an MPA length associated with a previous DDP segment:

1) is conducted in the case that the DDP header is referred by an MPA marker; and

2) is conducted in the case that the DDP header is not referred by the MPA marker.

25. The NIC of claim 21, wherein the conducting means conducts ~~c)~~, 1) in the case that the data transfer includes multiple DDP segments and a last DDP segment extends outside of the TCP segment boundary;

and conducts 2) in the case that the data transfer includes multiple DDP segments and a last DDP segment does not extend outside of the TCP segment boundary.

26. The NIC of claim 21, wherein the conducting means conducts ~~c),~~ 2) in the case that the data transfer includes a single DDP segment and an MPA length associated with the single DDP segment is greater than a transmission control protocol (TCP) segment length of the data transfer.

27. The NIC of claim 21, wherein the conducting means conducts 3) in the case that the data transfer includes a single DDP segment that has: an MPA length associated with the single DDP segment that equals a TCP segment length, and a valid CRC.

28. The NIC of claim 21, wherein the conducting means conducts 1) in the case that the data transfer includes a single DDP segment that has: an MPA length associated therewith that equals a TCP segment length, an invalid CRC and has a DDP header that is referred by an MPA length associated with a previous DDP segment.

29. The NIC of claim 28, wherein in the case that the single DDP segment that has: an MPA length associated therewith that equals a TCP segment length, an invalid CRC, and a DDP

header that is not referred by an MPA marker, the conducting means conducts:

1) in the case that the DDP header is referred by an MPA marker; and

2) in the case that the DDP header is not referred by an MPA marker.

30. The NIC of claim 19, further comprising means for setting the data transfer type to the first type when the conducting means conducts 2).

31. The NIC of claim 19, further comprising means for clearing TCP hole information in a connection context and stopping receipt reporting for an out-of-order data transfer upon which the means for conducting conducts 3).

32. The NIC of claim 19, wherein the data transfer includes DDP segments, and the calculating means calculates a CRC for all DDP segments of the data transfer together.

33. The NIC of claim 19, wherein the data transfer does not contain an MPA marker.

34. The NIC of claim 19, further comprising:

third means for storing a number of retransmission attempts for each data transfer including an error; and

fourth means for storing a largest sequence number.

35. The NIC of claim 34, wherein in the case that CRC is invalid for the data transfer, which indicates the data transfer is a newly received error-including data transfer:

the conducting means conducts 2) on the newly received error-including data transfer in the case that the number of retransmission attempts exceeds a maximum retransmission attempt number for that data transfer, and

the conducting means conducts 1) on the newly received error-including data transfer in the case that the number of retransmission attempts does not exceed a maximum retransmission attempt number for that data transfer; and

the NIC further comprising:

means for increasing the number of retransmission attempts for the newly received error-including data transfer by one in the case that the conducting means conducts 1); and

means for updating the fourth storing means to carry the largest sequence number among at least one previously received error-including data transfer and the newly received error-including data transfer in the case that the conducting means conducts ~~c),~~ 1).

36. The NIC of claim 34, further comprising:

means for resetting the number of retransmission attempts in the case that the CRC is valid for an in-order data transfer, and a sequence number of the in-order data transfer is greater than the stored largest sequence number; and

wherein the conducting means conducts 3) in the case that:

a) the CRC is valid for an in-order data transfer and the sequence number of the in-order data transfer is not greater than the stored largest sequence number, and

b) the resetting means resets the number of retransmission attempts.

37. A computer program product comprising a tangible computer useable medium having computer readable program code embodied therein, which, when executed by a computer infrastructure, enables the computer infrastructure to handle a data transfer in a network interface controller (NIC), the program product comprising:

program code configured to receive the data transfer wherein the data transfer is denoted as one of a first type and a second type;

program code configured to calculate a cyclical redundancy check (CRC) for the data transfer, wherein the CRC is one of valid and invalid;

program code configured to conduct, based on a comparison between a transfer control protocol (TCP) segment length and a marker with protocol data unit alignment (MPA) length and validity of a CRC of a direct data placement (DDP) segment within the data transfer, one of:

1) dropping the data transfer and not confirming reception;

2) placing the data transfer to a reassembly buffer of the NIC; and

3) placing the data transfer to an internal buffer of the NIC for direct data placement to a destination buffer.

38. The program product of claim 37, further comprising program code configured to set the data transfer type to the first type when the conducting program code conducts 2).

39. The program product of claim 37, further comprising program code configured to clear TCP hole information in a connection context and stop receipt reporting for an out-of-order data transfer upon which the conducting program code conducts 3).

40. The program product of claim 37, wherein the conducting program code conducts 2) in the case that the data transfer is of the first type.